A MULTISCALE SUB-LINEAR TIME FOURIER ALGORITHM FOR NOISY DATA

ANDREW CHRISTLIEB, DAVID LAWLOR, AND YANG WANG

Abstract. We extend the recent sparse Fourier algorithm of [LWC13] to the noisy setting. We present two such extensions, the second of which exhibits a novel form of error-correction not unlike that of the β-encoders in analog-to-digital conversion [DDGV06]. The algorithm runs in time $\mathrm{O}(k\log(k)\log(N/k))$ on average, provided the noise is not overwhelming. The error-correction property allows the algorithm to outperform FFTW over a wide range of sparsity and noise values, and is to the best of our knowledge novel in the sparse Fourier transform context.

1. Introduction

The Fast Fourier Transform (FFT) [CT65] is a fundamental numerical algorithm whose importance in a wide variety of applications cannot be overstated. The FFT reduces the runtime complexity of calculating the discrete Fourier transform (DFT) of a length N array from the naive $\mathrm{O}(N^2)$ to $\mathrm{O}(N\log(N))$. At the time of its introduction in the mid-1960s, it dramatically increased the size of problems that a typical computer could handle. Over the past fifty years the typical size of data sets has grown by orders of magnitude, and in certain application areas (e.g. ultra-wideband radar) the computation of the full FFT is no longer tractable on commodity hardware. In this and other instances, however, it is known *a priori* that the signals of interest have small frequency support; that is, their Fourier transforms are *sparse*. This problem has received attention from a number of research communities over the past decade, who have shown that it is possible to significantly outperform the FFT in both runtime and sampling requirements when the number of significant Fourier modes k is much less than the nominal bandwidth N.

The earliest work to specifically address the sparse Fourier transform problem was [GGI+02], which gave a randomized algorithm with runtime and sampling complexity $\mathrm{O}(k^2\,\mathrm{polylog}(N))$. This was later improved to $\mathrm{O}(k\,\mathrm{polylog}(N))$ [GMS05] through the use of unequally-spaced FFTs [AD96]. For a given failure probability δ and accuracy parameter ε, the algorithm

returns a k-term approximation $\hat{y}$ to the DFT of the input $\hat{x}$ that is near-optimal: with probabilty $1-\delta$ it holds that

$$\|\hat{x}-\hat{y}\|_2^2 \leq (1+\varepsilon)\|\hat{x}-\hat{x}_k\|_2^2. \tag{1.1}$$

Here $\hat{x}_k$ is the best k-term approximation to $\hat{x}$. A separate group of authors [HIKP12b] has developed a modified version of this algorithm with runtime $\mathrm{O}(\log(N)\sqrt{Nk\log(N)})$. While the dependence on N is sub-optimal asymptotically, in practice this algorithm is significantly faster than either [GGI+02] or [GMS05]. The same authors presented an improved algorithm with runtime $\mathrm{O}(k\log(N)\log(N/k))$ in [HIKP12a] whose frequency identification prodecure is very similar to [LWC13], upon which the present work is based. However, the performance of [HIKP12a] in the presence of noise has yet to be evaluated empirically.

The algorithms described in the previous paragraph are all randomized, and so will fail on some small subset of potential inputs. Recognizing this as a potential detriment in failure-intolerant applications, two authors have independently given deterministic algorithms for the sparse Fourier transform problem. In [Aka10] an algorithm with runtime $\mathrm{poly}(k,\log(N))$ was given where the exponent on k is at least six. This high dependence on k renders the algorithm infeasible in practice, and it has not been implemented. In [Iwe10], the combinatorial properties of aliasing among frequencies were exploited to give an algorithm with runtime and sampling complexity $\mathrm{O}(k^2\,\mathrm{polylog}(N))$. While this represented a major improvement over the theoretical runtime complexity of [Aka10], in practice it only outperformed the FFT for relatively modest values of the sparsity k.

Most recently the authors of [LWC13] gave a deterministic algorithm with average-case sampling and runtime complexity $\mathrm{O}(k\log(N))$. The worst-case runtime bounds are asymptotically of the same order as [Iwe10], but over a representative class of random signals it was shown to significantly outperform its deterministic and randomized competitors. This was achieved by sampling the input at two sets of equispaced points slightly offset in time. This time shift appears in the Fourier domain as a frequency modulation, which allows the authors to both detect when aliasing has occurred and, for frequencies that are isolated (i.e. not aliased), to calculate the frequency value directly. While [Iwe10] also uses properties of aliasing to reconstruct frequency values, it is not able to distinguish between aliased and non-aliased terms until sufficiently many DFTs of coprime lengths have been computed, and so is unable to perform any better in the average case than in the worst case.

In the empirical evaluation of [LWC13] an improvement of over two orders of magnitude was observed over [GMS05] and [Iwe10].

In this paper we extend the algorithm of [LWC13] to noisy environments in two distinct ways. The main result of this paper is a novel multiscale error-correcting algorithm that utilizes offset time samples at geometrically spaced shifts. This extension is in essence a progressive frequency identification algorithm not unlike the β-encoders for analog-to-digital conversion [DDGV06]. The new algorithm gives excellent performance in the noisy setting without significantly increasing the computational costs from the noiseless case. For comparison purposes we also present another extension, which is a minor modification of the noiseless algorithm based on a certain rounding of the frequency estimates. For both extensions we provide detailed mathematical analysis as well as empirical evaluations. While both extensions work well in the noisy environment, the multiscale algorithm achieves comparable or superior accuracy at a significantly lower computational cost.

The remainder of this paper is organized as follows. In Section 2 we review the notation introduced in [LWC13] that will be necessary for the sequel. We also describe our noise model, discuss some of the problems noisy signals present for the algorithm of [LWC13], and argue that in certain applications the ℓ_2 error metric is inappropriate and should be replaced with a form of Earth Mover's Distance. We also describe the random signal model used in the empirical evaluations in Sections 3 and 4. In Section 3 we give our first modified algorithm and analyze the dependence of the sampling rate on the noise level. We also illustrate its performance with an empirical evaluation. In Section 4 we describe our multiscale frequency identification procedure and provide an empirical evaluation. Finally in Section 5 we provide a brief conclusion.

2. Preliminaries

2.1. **Notation and brief review.** In this section we introduce the notation that will be used in the remainder of this paper and briefly review the results in [LWC13]. We consider frequency-sparse band-limited signals $S : [0, 1) \to \mathbb{C}$ of the form

$$S(t) = \sum_{\omega \in \Omega} a_\omega \mathrm{e}^{2\pi \mathrm{i} \omega t}, \tag{2.1}$$

where Ω is a finite set of integers bounded in $[-N/2, N/2)$ and $0 \neq a_\omega \in \mathbb{C}$ for each $\omega \in \Omega$. For simplicity we shall extend $S(t)$ periodically to a function on the whole real line. The Fourier samples of S are given by

$$\widehat{S}(h) = \int_0^1 S(t)\mathrm{e}^{-2\pi\imath ht}\,\mathrm{d}t, \ h \in \mathbb{Z}, \tag{2.2}$$

so that for signals of the form (2.1) we have $\widehat{S}(\omega) = a_\omega$ for $\omega \in \Omega$ and $\widehat{S}(h) = 0$ for all other $h \in \mathbb{Z}$. In practice we work with data of finite length. Given any finite sequence $\mathbf{s} = (s_0, s_1, \ldots, s_{p-1})$ of length p its discrete Fourier transform (DFT) is given by

$$\widehat{\mathbf{s}}[h] \ = \ \sum_{j=0}^{p-1} s_j \mathrm{e}^{-2\pi\imath jh/p} \ = \ \sum_{j=0}^{p-1} \mathbf{s}[j] W_p^{jh}, \tag{2.3}$$

where $h = 0, 1, \ldots, p-1$, $\mathbf{s}[j] := s_j$ and $W_p := \mathrm{e}^{-2\pi\imath/p}$ is the primitive p-th root of unity. The Fast Fourier Transform (FFT) [CT65] allows the computation of $\widehat{\mathbf{s}}$ in $\mathrm{O}(p \log p)$ steps.

All fast reconstruction algorithms apply the DFT to selected finite sample sets of $S(t)$, and our work is no exception. Let p be a positive integer and $\varepsilon > 0$. The two sample sets we use extensively are $\mathbf{S}_p$ and $\mathbf{S}_{p,\varepsilon}$, which are length p samples of $S(t)$ given by

$$\mathbf{S}_p[j] = S\Big(\frac{j}{p}\Big), \quad \mathbf{S}_{p,\varepsilon}[j] = S\Big(\frac{j}{p} + \varepsilon\Big), \quad j = 0, 1, \ldots, p-1.$$

For each h let $\Lambda_{p,h} = \{\omega \in \Omega : \omega \equiv h \pmod p\}$. It is a simple derivation to obtain

$$\widehat{\mathbf{S}}_p[h] = p \sum_{\omega \in \Lambda_{p,h}} a_\omega, \quad \widehat{\mathbf{S}}_{p,\varepsilon}[h] = p \sum_{\omega \in \Lambda_{p,h}} a_\omega \mathrm{e}^{2\pi\imath\varepsilon\omega}.$$

In the ideal scenario where all $\{\omega \pmod p : \ \omega \in \Omega\}$ are distinct we have

$$\widehat{\mathbf{S}}_p[h] = \begin{cases} pa_\omega & h \equiv \omega \pmod p \text{ for some } \omega \in \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

and similarly

$$\widehat{\mathbf{S}}_{p,\varepsilon}[h] = \begin{cases} pa_\omega \mathrm{e}^{2\pi\imath\varepsilon\omega} & h \equiv \omega \pmod p \text{ for some } \omega \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the nonzero elements of $\widehat{\mathbf{S}}_p[h]$ occur precisely at the locations $h = \omega \pmod p$ for some $\omega \in \Omega$, and moreover for such h we have $|\widehat{\mathbf{S}}_p[h]| = |\widehat{\mathbf{S}}_{p,\varepsilon}[h]|$. Furthermore for each $\omega \in \Omega$ and $h = \omega \pmod p$ we have $\frac{\widehat{\mathbf{S}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}_p[h]} = \mathrm{e}^{2\pi\imath\varepsilon\omega}$. Hence

$$2\pi\varepsilon\omega \equiv \mathrm{Arg}\left(\frac{\widehat{\mathbf{S}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}_p[h]}\right) \pmod{2\pi}, \tag{2.4}$$

where $\mathrm{Arg}(z)$ denotes the phase angle of the complex number z in $[-\pi, \pi)$. Now assume that we have $|\varepsilon| \leq \frac{1}{N}$. Then ω is completely determined by (2.4), as there will be no wrap-around aliasing. Hence

$$\omega = \frac{1}{2\pi\varepsilon} \mathrm{Arg}\left(\frac{\widehat{\mathbf{S}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}_p[h]}\right). \tag{2.5}$$

The weight a_ω can be recovered via $a_\omega = \widehat{\mathbf{S}}_p[h]/p$.

Remark. In fact, more generally, if we have an estimate of $\omega \in \Omega$, say $|\omega| < \frac{L}{2}$, then by taking $\varepsilon \leq \frac{1}{L}$ the same reconstruction formula (2.5) holds. We will use this observation in Section 4 when we develop a multiscale frequency identification procedure for noisy signals.

Of course it is possible that not all $\{\omega \pmod p : \ \omega \in \Omega\}$ are distinct. For an $\omega \in \Omega$ we say ω *has a collision modulo* p, or simply *has a collision* when there is no ambiguity, if there is at least one other $\omega' \in \Omega$ such that $\omega \equiv \omega' \pmod p$. In [LWC13] a criterion is developed to detect collisions in the noiseless case. For $\omega \in \Omega$ and $h = \omega \pmod p$, it is clear that a necessary condition for no collision to occur is

$$\left|\frac{\widehat{\mathbf{S}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}_p[h]}\right| = |\mathrm{e}^{2\pi\imath\varepsilon\omega}| = 1. \tag{2.6}$$

It is shown in [LWC13] that for a randomly chosen $\varepsilon > 0$ the converse holds with probability one, and furthermore checking the condition (2.6) for several ε would be sufficient to deterministically decide whether ω has a collision. In section 4 we use this latter observation to devise a robust test for collisions even in the presence of noise.

The algorithm developed in [LWC13] for recovering $S(t)$ is as follows: First we pick a prime $p = p_1$, which is roughly $5k$ where $k = |\Omega|$ is the number of modes in $S(t)$ (k is commonly referred to as the *sparsity* of $S(t)$). By taking p to be about 5 times the sparsity we ensure that on average collisions do not occur for more than 90% of $\omega \in \Omega$. Let Ω' denote the subset of Ω consisting of all non-collision $\omega \in \Omega$. For each $\omega \in \Omega'$ we recover $a_\omega \mathrm{e}^{2\pi\imath\omega t}$, and update $S(t)$ to

$$S_1(t) = S(t) - \sum_{\omega\in\Omega'} a_\omega \mathrm{e}^{2\pi\imath\omega t}.$$

We now apply the above procedure again for $S_1(t)$ with a different prime $p = p_2$ approximately in the range of $5k_1$, where $k_1 = k - |\Omega'|$ is now the sparsity for $S_1(t)$. This process is repeated until all modes are found.

In the implementation of the algorithm we set a small threshold in (2.6) to check for collisions. This means there is a small probability that a collision is undetected by our criterion and a false value ω_0 is put into Ω' when it shouldn't be. However no irreversible harm is done here because all this does is to create a new mode $-c_0 \mathrm{e}^{2\pi \mathrm{i} \omega_0 t}$ for some $c_0 \in \mathbb{C}$ in $S_1(t)$. By the use of different primes p_j in each iteration this false mode will eventually be identified and subtracted from the final reconstruction.

2.2. **Noise model.** In practice the samples we collect will be corrupted by some noise. In this paper we assume an i.i.d. noise model

$$\mathbf{S}^{\mathrm{n}}_p[j] = S\left(\frac{j}{p}\right) + \mathrm{n}_j = \mathbf{S}_p[j] + \mathrm{n}_j, \tag{2.7}$$

where $\mathrm{n} = (\mathrm{n}_j)$ are i.i.d. complex random variables with mean 0 and variance σ^2. A typical model is to assume $\{\mathrm{n}_j\}$ are i.i.d. complex Gaussian. In real world applications it is perhaps more realistic to set a bound on n_j, say $|\mathrm{n}_j| \le \tau$ for all j for some $\tau > 0$. In our numerical experiments we use Gaussian noise with a cut-off $\tau = 2\sigma$. With the noise model we have

$$\widehat{\mathbf{S}}^{\mathrm{n}}_p[h] = \widehat{\mathbf{S}}_p[h] + \widehat{\mathrm{n}}[h] \qquad \text{where} \qquad \widehat{\mathrm{n}}[h] = \sum_{j=0}^{p-1} \mathrm{n}_j \mathrm{e}^{-2\pi \mathrm{i} hj/p}. \tag{2.8}$$

By the i.i.d. property for $\{\mathrm{n}_j\}$

$$\mathbb{E}\big[\widehat{\mathrm{n}}[h]\big] = 0 \qquad \text{and} \qquad \mathrm{Var}\big[\widehat{\mathrm{n}}[h]\big] = p\sigma^2.$$

This yields

$$\mathbb{E}\left[\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]\right] = \widehat{\mathbf{S}}_p[h] \qquad \text{and} \qquad \mathbb{E}\left[|\widehat{\mathbf{S}}^{\mathrm{n}}_p[h] - \widehat{\mathbf{S}}_p[h]|^2\right] = p\sigma^2. \tag{2.9}$$

Thus, a typical noisy DFT coefficient $\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]$ will deviate from the true value $\widehat{\mathbf{S}}_p[h]$ by an amount proportional to $\sigma\sqrt{p}$. Similarly, for $\mathbf{S}^{\mathrm{n}}_{p,\varepsilon} = \mathbf{S}_{p,\varepsilon} + \mathrm{n}_\varepsilon$ we will have

$$\mathbb{E}\left[\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]\right] = \widehat{\mathbf{S}}_{p,\varepsilon}[h] \qquad \text{and} \qquad \mathbb{E}\left[|\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h] - \widehat{\mathbf{S}}_{p,\varepsilon}[h]|^2\right] = p\sigma^2. \tag{2.10}$$

We now pick a non-collision $\omega \in \Omega$. Then for $h = \omega \pmod p$ we will have

$$\begin{aligned} \widehat{\mathbf{S}}^{\mathrm{n}}_p[h] &= p a_\omega + \mathrm{O}(\sqrt{p}\sigma), \\ \widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h] &= p a_\omega \mathrm{e}^{2\pi \mathrm{i} \omega \varepsilon} + \mathrm{O}(\sqrt{p}\sigma). \end{aligned}$$

As a result a_ω can now be estimated easily via

$$a_\omega = \frac{1}{p} \widehat{\mathbf{S}}^{\mathrm{n}}_p[h] + \mathrm{O}\Big(\frac{\sigma}{\sqrt{p}}\Big). \tag{2.11}$$

The real challenge lies in the recovery of the frequencies in Ω. Assume that $|\widehat{\mathbf{S}}_{p,\varepsilon}|$ has a pulse at h. Then $h = \omega \pmod p$ for some $\omega \in \Omega$. If there is no collision for ω, in the noiseless environment ω is recovered via (2.5) as long as $\varepsilon < 1/N$. In the noisy setting $\widehat{\mathbf{S}}_{p,\varepsilon}[h]/\widehat{\mathbf{S}}_p[h]$ must be replaced by $\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]/\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]$. Interestingly, the mean of $\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]/\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]$ is in general *not* $\widehat{\mathbf{S}}_{p,\varepsilon}[h]/\widehat{\mathbf{S}}_p[h]$ as a result of the division. Nevertheless we have

$$\begin{aligned}\frac{\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]} &= \frac{\widehat{\mathbf{S}}_p[h]\mathrm{e}^{2\pi\imath\omega\varepsilon} + \widehat{\mathrm{n}}_\varepsilon[h]}{\widehat{\mathbf{S}}_p[h] + \widehat{\mathrm{n}}[h]} \\ &= \frac{\widehat{\mathbf{S}}_p[h]\mathrm{e}^{2\pi\imath\omega\varepsilon} + \mathrm{O}\left(\sigma\sqrt{p}\right)}{\widehat{\mathbf{S}}_p[h] + \mathrm{O}\left(\sigma\sqrt{p}\right)} \\ &= \frac{\mathrm{e}^{2\pi\imath\omega\varepsilon} + \mathrm{O}\left(\sigma/a_\omega\sqrt{p}\right)}{1 + \mathrm{O}\left(\sigma/a_\omega\sqrt{p}\right)} \\ &= \mathrm{e}^{2\pi\imath\omega\varepsilon} + \mathrm{O}\left(\sigma/a_\omega\sqrt{p}\right). \end{aligned} \tag{2.12}$$

Thus the ratio of noisy DFT coefficients agrees with the noiseless ratio up to an error term on the order of $\sigma/|a_\omega|\sqrt{p}$.

Given this estimate for the ratio of noisy DFT coefficients, we can derive bounds for the error in the Lee norm for the phase angle computed via $\mathrm{Arg}(z)$. Let $\mathcal{L}$ be a lattice in $\mathbb{R}$. For any $\theta \in \mathbb{R}$ the *Lee norm associated with the lattice* $\mathcal{L}$ for θ is given by the distance of θ to the lattice $\mathcal{L}$, i.e. $\|\theta\|_{\mathcal{L}} := \min_{k\in\mathcal{L}} |\theta - k|$. Under the Lee norm associated with the lattice $2\pi\mathbb{Z}$ it is well known that

$$\| \mathrm{Arg}\,(z+\eta) - \mathrm{Arg}(z)\|_{2\pi\mathbb{Z}} = \| \mathrm{Arg}\left(1 + z^{-1}\eta\right) \|_{2\pi\mathbb{Z}} \leq |z^{-1}\eta|. \tag{2.13}$$

Thus for a non-collision $\omega \in \Omega$ and $h = \omega \pmod p$, the estimates (2.13) and (2.12) combined yield

$$\left\| \mathrm{Arg}\left(\frac{\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]}\right) - 2\pi\omega\varepsilon \right\|_{2\pi\mathbb{Z}} \leq \mathrm{O}\left(\frac{\sigma}{|a_\omega|\sqrt{p}}\right). \tag{2.14}$$

When we apply the estimate (2.5) for ω under the noise model we will end up with an approximation

$$\omega^{\mathrm{n}} := \frac{1}{2\pi\varepsilon}\,\mathrm{Arg}\left(\frac{\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}^{\mathrm{n}}_p[h]}\right)$$

such that

$$\|\omega^{\mathrm{n}} - \omega\|_{N\mathbb{Z}} \leq \mathrm{O}\left(\frac{\sigma}{2\pi\varepsilon|a_\omega|\sqrt{p}}\right). \tag{2.15}$$

Now if we apply the algorithm developed in [LWC13] the ratio $\sigma/\varepsilon\sqrt{p}$ is critical in determining the sensitivity of our phase estimation (as well as the weight estimation) to noise. Without any modifications to the algorithm it is thus important that we choose the lengths p so that $\sigma/\varepsilon\sqrt{p}$ is within the tolerance.

2.3. **Earth mover distance.** In the existing literature on the sparse Fourier transform, the ℓ_2 norm is most often used to assess the quality of approximation. There are many reasons for this choice, with the two most convincing perhaps being the completeness of the complex exponentials with respect to the ℓ_2 norm and Parseval's theorem. For certain applications, however, this choice of norm is inappropriate. For example, in wide-band spectral estimation and radar applications, one is interested in identifying a set of frequency intervals containing active Fourier modes. In this case, an estimate $\widetilde{\omega}$ of the true frequency ω with $|\widetilde{\omega}-\omega| \ll N$ is useful, but unless $\widetilde{\omega}=\omega$ the ℓ_2 metric will report an O(1) error. For these reasons, we propose measuring the approximation error of sparse Fourier transform problems with the Earth Mover Distance (EMD) [RTG00]. Originally developed in the context of content-based image retrieval, EMD measures the minimum cost that must be paid (with a user-specified cost function) to transform one distribution of points into another. EMD can be calculated efficiently as the solution of a linear program corresponding to a certain flow minimization problem.

For our problem, we consider the cost to move a set of estimated Fourier modes and coefficients $\big\{(\widetilde{\omega}_j, c_{\widetilde{\omega}_j})\big\}_{j=1}^{\widetilde{k}}$ to the true values $\big\{(\omega_j, c_{\omega_j})\big\}_{j=1}^{k}$ under the cost function

$$d_1\big((\omega, c_\omega), (\widetilde{\omega}, c_{\widetilde{\omega}}); N\big) := \frac{|\omega-\widetilde{\omega}|}{N} + |c_\omega - c_{\widetilde{\omega}}|.$$

This choice of cost function strikes a balance between the fidelity of the frequency estimate (as a fraction of the bandwidth) and that of the coefficient estimate. We also consider the "phase-only" cost function

$$d_\omega(\omega_1, \omega_2; N) := \frac{|\omega_1-\omega_2|}{N},$$

which provides a measure of how close our frequency estimates are to the true values. We denote the EMD using d_1 by EMD(1) and using d_ω by EMD(ω) in our empirical studies in Subsections 3.2 and 4.4 below.

2.4. **Random signal model.** For the empirical evaluations in Subsections 3.2 and 4.4 we consider test signals with uniformly random phase over the bandwidth and coefficients

chosen uniformly from the complex unit circle. In other words, given k and N, we choose k frequencies ω_j uniformly at random (without replacement) from $[-N/2, N/2) \cap \mathbb{Z}$. The corresponding Fourier coefficients a_j are of the form $\mathrm{e}^{2\pi\imath\theta_j}$, where θ_j is drawn uniformly from $[0, 1)$. The signal is then given by

$$S(t) = \sum_{j=1}^{k} a_j \mathrm{e}^{2\pi\imath\omega_j t}. \tag{2.16}$$

This is the standard signal model considered in previous empirical evaluations of sub-linear Fourier algorithms [IGS07, Iwe10, HIKP12b, LWC13].

3. Rounding: A Minor Modification of Noiseless Algorithm

A simple modification to the noiseless algorithm of [LWC13] for the noisy case is to increase the sample lengths p. By choosing p large enough the error from noise can be mitigated to be within a given tolerance. The modification can be viewed simply as rounding, and we include it both as a more direct and simple to implement extension as well as for comparison purposes. When the noise level is low, this modification yields reasonably good results.

As in the noiseless case we choose the shift $\varepsilon > 0$ so that $\varepsilon \leq 1/N$. In the noiseless case $\varepsilon = 1/N$ would be sufficient to avoid wrap-around aliasing in the phase reconstruction. Due to the presence of noise we will need to make ε slightly smaller because of (2.15). Let us analyze the recovery process for an $\omega \in \Omega$ if we simply carry out the same process as in the noiseless environment.

First we choose a length p. Assume that $\omega \in \Omega$ does not collide with any other $\omega' \in \Omega$ modulo p. Let $h = \omega \pmod{p}$. The reconstruction of ω utilizes two factors. First, the location of peaks in the DFT are robust to noise: even with a relatively high noise level we may take $h = \omega \pmod{p}$ to be exact. Second, by (2.15) the frequency reconstruction from noisy measurements is correct up to an error term of size $\mathrm{O}\left(\sigma/2\pi\varepsilon|a_\omega|\sqrt{p}\right)$. By combining these two measures we can more reliably estimate ω.

Our proposed modification is to simply round the noisy frequency estimate

$$\widetilde{\omega} = \frac{1}{2\pi\varepsilon} \operatorname{Arg}\left(\frac{\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}^{\mathrm{n}}_{p}[h]}\right)$$

to the nearest integer of the form $np + h$. This improved estimate is therefore given by

$$\widetilde{\omega}' = p \cdot \text{round}\left(\frac{\widetilde{\omega} - h}{p}\right) + h, \tag{3.17}$$

where round(x) returns the nearest integer to x. For low noise levels this modification will return the true value ω, while for larger noise levels it is possible that $\widetilde{\omega}$ deviates by more than $p/2$ from the true frequency ω. In this case the estimate $\widetilde{\omega}'$ will be wrong by a multiple of p. Larger values of p will reduce the likelihood of an error in frequency estimation.

To ensure that the estimated frequencies are sufficiently far from the branch cut of $\text{Arg}(z)$ along the negative real axis, we take the shift $\varepsilon \leq 1/2N$. The estimated frequencies then satisfy $-N \leq \widetilde{\omega} < N$, while the true frequencies lie in the smaller interval $[-N/2, N/2)$. It is thus extremely unlikely that the deviations due to the noise will push the estimates across the discontinuity.

We saw in the previous section that the error in the phase estimation is on the order of $\sigma p^{-1/2}$ when using the reconstruction formula (2.5). When using the rounding procedure (3.17), however, we should expect accurate results for a wider range of sample lengths p and noise levels σ. Indeed, note that the rounded frequency estimate $\widetilde{\omega}'$ is *exact* as long as

$$|\widetilde{\omega} - \omega| < \frac{p}{2}. \tag{3.18}$$

Recall from Section 2.2 that the error of the frequency estimate $\widetilde{\omega}$ is on the order of $\text{O}(\sigma/\varepsilon\sqrt{p})$. Let us assume that it is bounded by $C\sigma/\varepsilon\sqrt{p}$ for some constant C. Combining this with the requirement (3.18) we see that the rounded frequency estimate $\widetilde{\omega}'$ will be exact provided

$$C\frac{\sigma}{\varepsilon p^{3/2}} < \frac{1}{2}. \tag{3.19}$$

It follows that we get exact reconstruction if $p \geq (2C\sigma/\varepsilon)^{2/3}$.

To illustrate this relationship, we generated 1000 test signals with frequencies chosen uniformly at random from $[-N/2, N/2)$ and set the corresponding coefficient to unity. Thus our test signals for this empirical trial were one-term trigonometric polynomials. For this test we took $N = 2^{22}$ and investigated a range of parameters (σ, p). We reconstructed the frequencies in two ways: first, simply using the formula (2.5), and second by combining this estimate with the rounding procedure (3.17). In Figure 1 we plot the average phase error

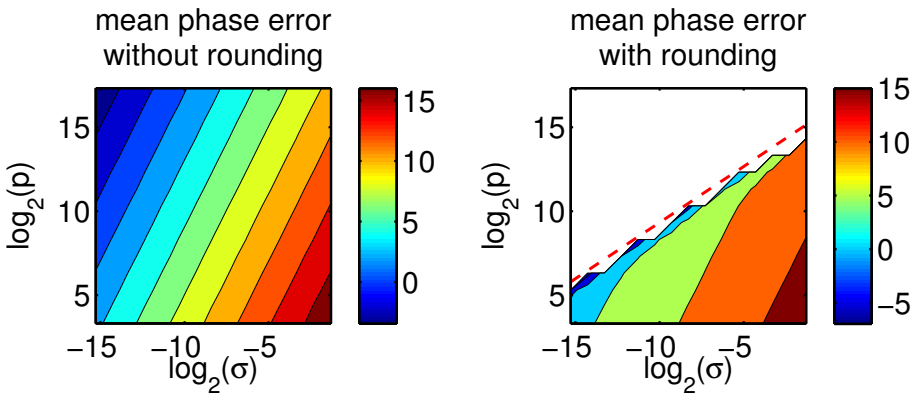


FIGURE 1. (left) Mean phase error (in log scale) for frequency estimation via (2.5). (right) Mean phase error (in log scale) for frequency estimation with rounding via (3.17). The red dotted line marks the transition to exact recovery when $p > (2\sigma/\varepsilon)^{2/3}$.

in logarithmic scale as a function of both σ and p, which were varied from 2.5×10^{-5} to 0.4906 and from 10 to 163840, respectively, by powers of two.

In the plot on the left, which corresponds to reconstruction using only (2.5), we can clearly see the contours of constant phase error obeying the relationship $\log_2(p) = 2\log_2(\sigma) + \alpha$ for various α. This confirms our analytic estimate from Subsection 2.2 that the phase error is proportional to $\sigma/\sqrt{p}$. In the plot on the right, which corresponds to the improved reconstruction using (3.17), we can see that for large values of σ and small values of p the same relationship holds. However, for smaller σ and larger p we see an abrupt transition to exact reconstruction (the white area in the upper-left). The boundary of this region (red dotted line) follows the relationship $\log_2(p) = \frac{2}{3}\log_2(\sigma) + 16$, corresponding to $C = 1$ in (3.19) above. This illustrates that for small enough values of the ratio $\sigma/\varepsilon p^{3/2}$ the rounding procedure is exact.

3.1. **Algorithm.** Our first algorithm for noisy signals is only a slight modification of the noiseless algorithm presented in [LWC13, Algorithm 1]. Considering (3.19), we change the lower bound

$$p > c_1 k \tag{3.20}$$

to

$$p > \max\{c_1 k, c_2(\sigma/\varepsilon)^{2/3}\}. \tag{3.21}$$

In this way we ensure that the choice of p is always large enough to isolate most of the k frequencies on average as well as being large enough to ensure that the rounding procedure (3.17) is exact. In all of our experiments below we took $c_2 = 4$.

3.2. **Empirical evaluation.** In this section we describe the results of an empirical evaluation of the algorithm with the rounding procedure (3.17) and new choice of sampling rate p given by (3.21). We focus on two aspects of the algorithm's performance: accuracy as measured in the EMD(1) metric (c.f. Subsection 2.3), and runtime as a function of both the sparsity k and the noise level σ. In all of the experiments reported below, we report averages over 100 random test signals generated according to the prescription in Subsection 2.4. The bandwidth for these test was fixed at $N = 2^{22}$, and the shift ε was taken to be $1/2N$. All experiments were conducted in C++ on a Linux machine with four Intel Xeon X5355 dual-core processors at 2.66 GHz and 64 Gb of RAM. The Intel compiler was used with optimization flag -O3. All FFTs are performed using FFTW3 [FJ05], and timing results are reported in CPU ticks using the "cycle.h" header included with the FFTW3 source code.

3.2.1. *Accuracy.* In Figure 2 we plot the average EMD(1) error of the algorithm with the rounding modification as a function of the noise level σ. It is clear that the error increases linearly with the noise level, which indicates that the method is robust with respect to additive noise in the signal. Moreover, the EMD(ω) error was zero for all trials of the rounding algorithm, confirming that taking $p > c_2(\sigma/\varepsilon)^{2/3}$ suffices to exactly recover the true frequencies.

3.2.2. *Runtime.* In Figure 3 (A) we plot the average runtime (in CPU ticks) as a function of the sparsity k for a fixed value of the noise level $\sigma = 0.512$ and the parameter $c_1 = 2$. Note that the Fourier coefficients in our test signals all have unit magnitude, so this is a severely disadvantageous situation. For this high level of noise, we see that there is no dependence on k until $k = 64$; this is a consequence of the requirement (3.21) on the choice of sampling rate. As a reference for runtime comparisons, FFTW3 takes approximately 10^9 CPU ticks on the same machine. Thus at this noise level our modified algorithm is slightly slower than a highly optimized FFT implementation.

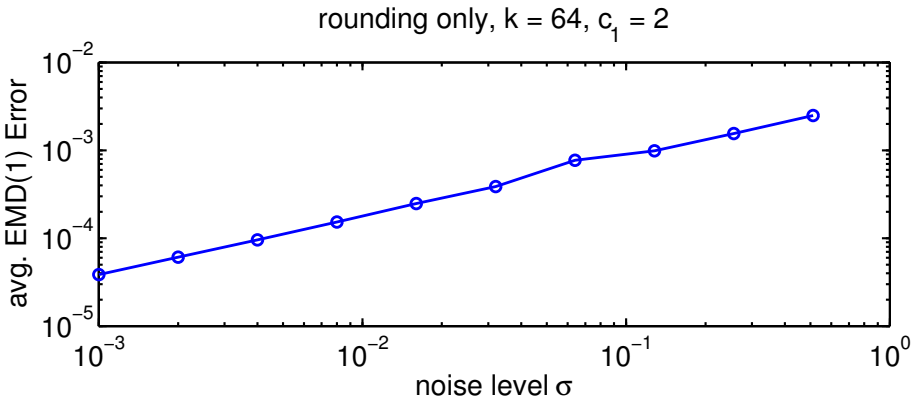
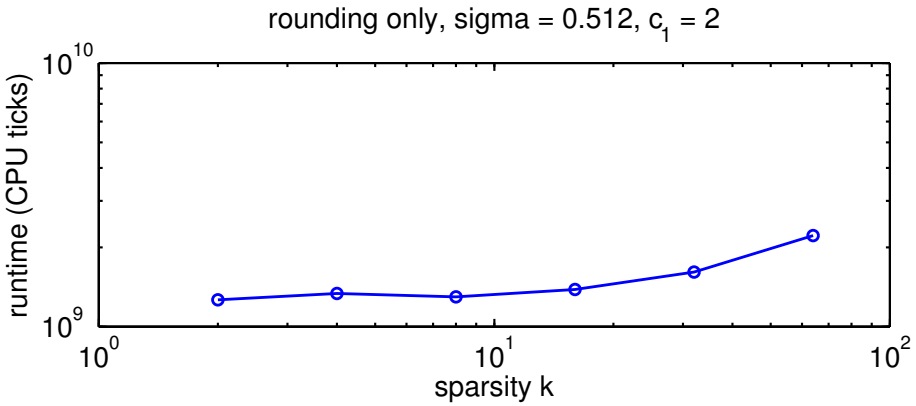


FIGURE 2. Average EMD(1) error of the algorithm with rounding modification as a function of noise level σ. The EMD(ω) error was zero for all runs.

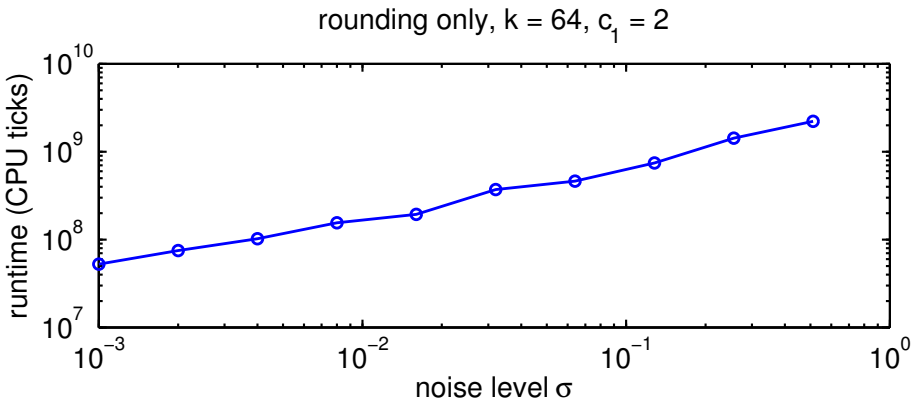
In Figure 3 (B) we plot the average runtime (in CPU ticks) as a function of the noise level σ for a fixed value of the sparsity $k = 64$. From the figure we can see the approximate dependence of the runtime on $\sigma^{2/3}$, as dictated by the choice of p in (3.21). Moreover, for $\sigma \leq 0.032$, the algorithm is faster than FFTW. In the next section we develop a more sophisticated algorithm which outperforms FFTW for a wider range of noise levels.

4. A Multiscale Algorithm

In Section 3 we saw that taking $p > \max\{c_1 k, c_2(\sigma/\varepsilon)^{2/3}\}$ sufficed to ensure that the rounding procedure was exact. While this gives good results in terms of accuracy, the increased runtime associated with larger noise levels is undesirable. The main contribution of this paper is a multiscale algorithm for recovering the frequency set Ω of the signal $S(t)$. This algorithm achieves equal if not superior accuracy while providing improvement by several orders of magnitude in computational efficiency. The key feature of this multiscale algorithm is the employment of multiple shifts ε_j, which enable us to improve the accuracy of the phase estimations progressively without the need to significantly increase the sample length p. In Subsection 4.1 we give some background on our multi-scale method and introduce the main idea of our algorithm. In Subsection 4.2 we prove that our multiscale approximations are accurate estimates of the true frequencies. In Subsection 4.3 we describe the basic multi-scale algorithm, and in Subsection 4.4 we report the results of an empirical evaluation.



(A)



(B)

FIGURE 3. (A) Average runtime (in CPU ticks) vs. sparsity k for the algorithm with rounding modification. (B) Average runtime vs. noise level σ.

4.1. **Multiscale frequency estimation.** The main idea for the multiscale algorithm is that a value can be estimated with high precision with an inaccurate (coarse) estimator applied progressively at different scales, much like in analog-to-digital conversion where a signal value can be estimated with very high precision by the very coarse binary quantization. In our sparse Fourier recovery algorithm, the coarse estimator is the approximation formula given by (2.14)

$$\varepsilon\omega =_{\mathbb{Z}} \frac{1}{2\pi} \operatorname{Arg}\left(\frac{\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon}[h]}{\widehat{\mathbf{S}}^{\mathrm{n}}_{p}[h]}\right), \tag{4.1}$$

where $=_{\mathbb{Z}}$ is measured by the Lee norm $\|\cdot\|_{\mathbb{Z}}$.

For simplicity let us assume for the moment that our signal contains a single frequency ω with non-zero Fourier coefficient. For a fixed p, let $\widetilde{\omega}$ be our estimate for ω using the rounding procedure from Section 3 with shift $\varepsilon_0 < 1/N$. Then we have

$$\widetilde{\omega} = \omega \pmod{p}, \tag{4.2}$$

although in general $\widetilde{\omega}$ may differ from ω by a multiple of p.

Suppose now that we repeat the computation of $\widetilde{\omega}$ using a larger shift $\varepsilon_1 > \varepsilon_0$; that is, we sample our signal at time points $j/p + \varepsilon_1$, take the FFT, and compute

$$b_1 = \frac{1}{2\pi} \operatorname{Arg}\left(\frac{\widehat{\mathbf{S}}^{\mathrm{n}}_{p,\varepsilon_1}[h]}{\widehat{\mathbf{S}}^{\mathrm{n}}_{p}[h]}\right) \tag{4.3}$$

(note that we do not divide by ε_1). Since in general $\varepsilon_1 > 1/N$, we cannot take b_1/ε_1 as an estimate for ω, although it still holds that

$$b_1 \approx \varepsilon_1\omega \pmod{[-\tfrac{1}{2},\tfrac{1}{2})}, \tag{4.4}$$

where $x \pmod{[-\frac{1}{2},\frac{1}{2})}$ is the unique value y in $[-\frac{1}{2},\frac{1}{2})$ such that $x \equiv y \pmod 1$. We can use this fact to estimate the error $\omega - \widetilde{\omega}$ as follows. Note that

$$\begin{aligned} \varepsilon_1(\omega - \widetilde{\omega}) &= \varepsilon_1\omega - \varepsilon_1\widetilde{\omega} \\ &\approx (b_1 - \varepsilon_1\widetilde{\omega}) \pmod{[-\tfrac{1}{2},\tfrac{1}{2})}, \end{aligned} \tag{4.5}$$

so that

$$\omega - \widetilde{\omega} \approx (b_1 - \varepsilon_1\widetilde{\omega}) \pmod{[-\tfrac{1}{2},\tfrac{1}{2})}/\varepsilon_1. \tag{4.6}$$

This estimate of the error is not exact, since there is still noise that can perturb the calculated value b_1 from the true value $\varepsilon_1\omega \pmod{[-\frac{1}{2},\frac{1}{2})}$. However, analogous to (2.15) we have

$$(\omega - \widetilde{\omega}) - (b_1 - \varepsilon_1\widetilde{\omega}) \pmod{[-\tfrac{1}{2},\tfrac{1}{2})}/\varepsilon_1 = \mathrm{O}\left(\frac{\sigma}{\varepsilon_1\sqrt{p}}\right),$$

which immediately implies that the updated estimate satisfies

$$\omega - \left(\widetilde{\omega} + (b_1 - \varepsilon_1\widetilde{\omega}) \pmod{[-\tfrac{1}{2},\tfrac{1}{2})}/\varepsilon_1\right) = \mathrm{O}\left(\frac{\sigma}{\varepsilon_1\sqrt{p}}\right). \tag{4.7}$$

Since $\varepsilon_1 > \varepsilon_0$, adding the correction term (4.6) to our previous estimate $\widetilde{\omega}$ will give a finer approximation to the true frequency ω. By iterating this error correction process with progressively larger shifts ε_j, we obtain an algorithm which adaptively corrects for the error

in a multiscale fashion. In the next section we provide a detailed analysis of this multiscale approximation scheme, and prove that the frequency estimates it produces are accurate.

4.2. **Analysis of multiscale approximations.** We begin with a technical lemma relating arithmetic in the Lee norm $\|\cdot\|_{\mathbb{Z}}$ to that on the interval $[-\frac{1}{2}, \frac{1}{2})$. It will be used repeatedly in the sequel.

Lemma 4.1. *Let $\delta > 0$ and $x \in [-\frac{1}{2}+\delta, \frac{1}{2}-\delta]$. Assume that $\|x-b\|_{\mathbb{Z}} < \delta$ and $b \in [-\frac{1}{2}, \frac{1}{2})$. Then $|x-b| < \delta$.*

Proof. Let $r = \|x-b\|_{\mathbb{Z}}$. Then $x - b = \pm r + k$ for some $k \in \mathbb{Z}$. If $k = 0$ we are done. Assume $k \neq 0$. Note that $|x-b| \leq |x| + |b| \leq 1 - \delta$. However, $|\pm r + k| \geq 1 - r > 1 - \delta$. This is a contradiction. ∎

The following theorem formalizes the multiscale frequency estimation procedure which was introduced in the previous subsection.

Theorem 4.2. *Let $\omega \in [-\frac{N}{2}, \frac{N}{2})$. Let $0 < \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_m$ and $b_0, b_1, \ldots, b_m \in \mathbb{R}$ such that*

$$\|\varepsilon_j \omega - b_j\|_{\mathbb{Z}} < \delta, \qquad 0 \leq j \leq m$$

where $0 < \delta \leq \frac{1}{4}$. Assume that $\varepsilon_0 \leq \frac{1-2\delta}{N}$ and $\beta_j := \varepsilon_j/\varepsilon_{j-1} \leq (1-2\delta)/(2\delta)$. Then there exist $c_0, c_1, \ldots, c_m \in \mathbb{R}$, each computable from $\{\varepsilon_j\}$ and $\{b_j\}$, such that

$$|\widetilde{\omega} - \omega| \leq \frac{\delta}{\varepsilon_0} \prod_{j=1}^{m} \beta_j^{-1}, \qquad \text{where} \qquad \widetilde{\omega} := \sum_{j=0}^{m} \frac{c_j}{\varepsilon_j}.$$

Proof. Denote $\omega_0 := \omega$. We first note that $|\varepsilon_0 \omega_0| \leq \varepsilon_0 \frac{N}{2} \leq \frac{1}{2} - \delta$. Let $c_0 = b_0 \pmod{[-\frac{1}{2}, \frac{1}{2})}$, so that $|\varepsilon_0 \omega_0 - c_0| < \delta$ by Lemma 4.1. Let $\lambda_0 = c_0/\varepsilon_0$, which represents a coarse estimate of ω_0 with the error bound $|\lambda_0 - \omega_0| < \delta/\varepsilon_0$.

Let $\omega_1 = \omega_0 - \lambda_0$. By the above $|\omega_1| < \delta/\varepsilon_0$ and

$$|\varepsilon_1 \omega_1| < \frac{\varepsilon_1 \delta}{\varepsilon_0} = \beta_1 \delta \leq \frac{1}{2} - \delta.$$

Now $\|\varepsilon_1 \omega - b_1\|_{\mathbb{Z}} = \|\varepsilon_1 \omega_1 - (b_1 - \varepsilon_1 \lambda_0)\|_{\mathbb{Z}} < \delta$. Set $c_1 = b_1 - \varepsilon_1 \lambda_0 \pmod{[-\frac{1}{2}, \frac{1}{2})}$. It follows from Lemma 4.1 again that $|\varepsilon_1 \omega_1 - c_1| < \delta$. We set $\lambda_1 = c_1/\varepsilon_1$.

We can recursively define c_j, λ_j and ω_j for all $1 \le j \le m$. In general we define $\omega_j := \omega_{j-1} - \lambda_{j-1}$. This leads to

$$|\varepsilon_j \omega_j| < \frac{\varepsilon_j \delta}{\varepsilon_{j-1}} = \beta_j \delta \le \frac{1}{2} - \delta.$$

Set $c_j = (b_j - \varepsilon_j \lambda_{j-1}) \pmod{[-\frac{1}{2}, \frac{1}{2})}$. Then $\|\varepsilon_j \omega_j - c_j\|_{\mathbb{Z}} < \delta$. Lemma 4.1 now yields $|\varepsilon_j \omega_j - c_j| < \delta$. Set $\lambda_j = c_j/\varepsilon_j$.

Finally denote $\omega_{m+1} = \omega_m - \lambda_m$. It is straightforward now to verify that

$$\omega = \omega_0 = \sum_{j=0}^{m} \lambda_j + \omega_{m+1} = \sum_{j=0}^{m} \frac{c_j}{\varepsilon_j} + \omega_{m+1}.$$

Furthermore, by construction $\omega_{m+1} = \omega_m - \lambda_m$, which has $|\omega_{m+1}| \le \delta/\varepsilon_m$. By hypothesis $\varepsilon_m = \varepsilon_0 \prod_{j=1}^{m} \beta_j$. Thus

$$|\omega_{m+1}| \le \frac{\delta}{\varepsilon_0} \prod_{j=1}^{m} \beta_j^{-1}.$$

The theorem is now proved. ∎

Remark 4.1. From the proof of Theorem 4.2 the values c_j and $\widetilde{\omega}$ are explicitly computable through the recursive formula $\omega_0 = \omega$, $c_0 = b_0 \pmod{[-\frac{1}{2}, \frac{1}{2})}$, $\lambda_0 = c_0/\varepsilon_0$ and

$$\left\{ \begin{array}{rcl} \omega_j & = & \omega_{j-1} - \lambda_{j-1} \\ c_j & = & (b_j - \varepsilon_j \lambda_{j-1}) \pmod{[-\frac{1}{2}, \frac{1}{2})} \\ \lambda_j & = & c_j/\varepsilon_j \end{array} \right. \tag{4.8}$$

for $1 \le j \le m$.

Corollary 4.3. *Assume that in the above theorem we have $\beta_j = \beta$ where $\beta \le (1-2\delta)/(2\delta)$, i.e. $\varepsilon_j = \beta^j \varepsilon_0$ for all j. Let $p > 0$ and $m \ge \left\lfloor \log_\beta \frac{2\delta}{p\varepsilon_0} \right\rfloor + 1$. Then*

$$|\widetilde{\omega} - \omega| \le \frac{\delta}{\varepsilon_0} \beta^{-m} < \frac{p}{2}.$$

Proof. This is a straightforward corollary. By Theorem 4.2 we have

$$|\widetilde{\omega} - \omega| \le \frac{\delta}{\varepsilon_0} \prod_{j=1}^{m} \beta_j^{-1} = \frac{\delta}{\varepsilon_0} \beta^{-m}.$$

It is easy to check that $m = \left\lfloor \log_\beta \frac{2\delta}{p\varepsilon_0} \right\rfloor + 1$ is the smallest integer such that $\frac{\delta}{\varepsilon_0} \beta^{-m} < \frac{p}{2}$. ∎

Note that as we have mentioned in Section 3, even with noise the value $\omega \pmod{p}$ can be precisely computed very reliably. Thus if the difference $|\omega - \widetilde{\omega}|$ is smaller than $\frac{p}{2}$ then ω can be recovered exactly by taking the closest integer to $\widetilde{\omega}$ with the same residue modulo p.

In numerical tests we usually choose uniform $\beta_j = \beta$. While making β as large as it can be for a given error estimate δ will undoubtedly reduce the computational cost, there is nevertheless a good reason that we should not be too "greedy" and be more conservative by choosing a smaller $\beta > 1$. The reason is that given the random nature of the noise the error bound δ is only in the average sense. To minimize reconstruction errors we should try to provide as much latitude as possible for the uncertainties associated with the error estimate δ. Hence it is useful to ask how much latitude does one get for given choices of ε_0 and β.

Theorem 4.4. *Let $\omega \in [-\frac{N}{2}, \frac{N}{2})$, $\varepsilon_0 > 0$ and $\beta > 1$. Set $\varepsilon_j = \beta^j \varepsilon_0$ for $1 \le j \le m$. Assume that we have $b_0, b_1, \ldots, b_m \in \mathbb{R}$ such that*

$$\|\varepsilon_j \omega - b_j\|_{\mathbb{Z}} < \delta, \qquad 1 \le j \le m$$

where

$$\delta = \min\Big(\frac{1 - \varepsilon_0 N}{2}, \frac{1}{2\beta + 2}\Big). \tag{4.9}$$

Then the estimate $\widetilde{\omega}$ of ω given by $\widetilde{\omega} := \sum_{j=0}^{m} \frac{c_j}{\varepsilon_j}$ satisfies

$$|\widetilde{\omega} - \omega| \le \frac{\delta}{\varepsilon_0} \beta^{-m},$$

where c_j are given in (4.8).

Proof. The proof is straightforward. Note that Theorem 4.2 holds under the conditions $\varepsilon_0 \le \frac{1-2\delta}{N}$ and $\beta_j \le \frac{1-2\delta}{2\delta}$. These two conditions are equivalent to the condition $\delta \le \min\big(\frac{1-\varepsilon_0 N}{2}, \frac{1}{2\beta+2}\big)$. Clearly, $\delta = \min\big(\frac{1-\varepsilon_0 N}{2}, \frac{1}{2\beta+2}\big)$ is the largest admissible value for δ. ∎

4.3. **Algorithm.** In this section we provide some details of our implementation of the multiscale frequency estimation procedure described in Subsection 4.1. In particular, we discuss the choice of various parameters necessary for reconstruction according to Theorem 4.2 as well as changes made to the aliasing detection test from [LWC13] to improve robustness in the presence of noise.

4.3.1. *Number of iterations.* Recall from Corollary 4.3 that, for constant $\beta_j = \beta$, $m = \Big\lfloor \log_\beta \frac{2\delta}{p\varepsilon_0} \Big\rfloor + 1$ shifts suffices to ensure that the estimated frequency satisfies $|\widetilde{\omega} - \omega| < \frac{p}{2}$. As in Section 3 we take $\varepsilon_0 = 1/2N$ to avoid the branch of $\mathrm{Arg}(z)$; furthermore, as in [LWC13] the sample lengths are p are chosen proportional to the sparsity k. Thus after $\mathrm{O}(\log(N/k))$

iterations, by rounding the approximate frequency $\widetilde{\omega}$ to the closest integer of the form $np+h$, where $h = \omega \pmod p$ is known from the location of the peak in $\widehat{\mathbf{S}}_p^{\mathrm{n}}$, we will recover the true frequency ω. With the results of [LWC13] this immediately implies that the average-case runtime of the multiscale algorithm is $\mathrm{O}(k\log(k)\log(N/k))$.

4.3.2. *Choice of β.* It remains to determine the choice of β, given the sample length p and the noise level σ. Recall from the proof of Theorem 4.2 that the estimated frequency $\widetilde{\omega}$ is given by the sum $\sum_{j=1}^{m}\lambda_j$, where $\lambda_j = c_j/\varepsilon_j$. Moreover, the difference between successive frequency approximations is given in terms of λ_j as

$$\omega_j := \omega_{j-1} - \lambda_{j-1} \quad \Longrightarrow \quad \lambda_j = \omega_j - \omega_{j+1}.$$

Thus we can decompose the error of approximation at stage $j+1$ as

$$\begin{aligned} |\omega - \omega_{j+1}| &= |(\omega_j - \omega_{j+1}) - (\omega_j - \omega)| \\ &= |\lambda_j - (\omega_j - \omega)|. \end{aligned} \tag{4.10}$$

By Theorem 4.2 the left-hand side of (4.10) satisfies

$$|\omega - \omega_{j+1}| < \frac{\delta}{\varepsilon_{j+1}},$$

while analogously to (2.15) the right-hand side of (4.10) satisfies

$$|\lambda_j - (\omega_j - \omega)| \leq \mathrm{O}\left(\frac{\sigma}{2\pi\varepsilon_j\sqrt{p}}\right).$$

Denoting by c_σ the constant in the right-hand side above and equating the two upper bounds gives

$$\frac{2\pi\delta\sqrt{p}}{c_\sigma\sigma} = \frac{\varepsilon_{j+1}}{\varepsilon_j} =: \beta. \tag{4.11}$$

As mentioned after Corollary 4.3, in our implementation we take β to be somewhat smaller than (4.11) would suggest, using the value $\beta = \frac{\sqrt{p}}{2c_\sigma\sigma}$. Note that for large σ and small p this quantity could be less than one. In order to ensure $\beta > 1$ we require the sample lengths to satisfy $p > \max\{c_1 k, (2c_\sigma\sigma)^2\}$; due to this the runtime bound in the previous subsection holds only for $\sigma \leq \frac{\sqrt{c_1 k}}{2c_\sigma}$.

4.3.3. *Robust aliasing test.* As noted in Subsection 2.1, our frequency estimation procedure works only for non-collsion ω. In [LWC13] two tests were given to determine whether a collision had occurred at a candidate frequency. In the implementation of that algorithm in

the noiseless setting, requiring the ratio (2.6) to be within some threshold of unity sufficed to detect collisions. In the setting of the current paper, where the samples are corrupted with noise, we resort to the second of the tests given in in [LWC13], which examines the ratios (2.6) for several values of ε. For $0 \leq j \leq m$ we compute the ratio (2.6) and compare it with a threshold τ. We count the number of times the ratio exceeds τ and reject those frequencies which fail more than an η fraction of the tests. Since we expect fluctuations in this ratio due to noise of order $\sigma/\sqrt{p}$ we set τ to be a small constant multiple of this quantity.

In Algorithm 1 we give pseudocode for the iterative frequency estimation procedure; the full algorithm is given by replacing the calculation of frequencies in [LWC13, Algorithm 1] with this procedure.

4.4. **Empirical evaluation.** In this section we present the results of an empirical evaluation of our multiscale algorithm. As in Subsection 3.2, we focus on accuracy and runtime, and report average quantities over 100 random test signals. The test signals were generated in the same manner as in Subsection 3.2 and were evaluated on the same machine. After extensive testing it was determined that the choice of parameters $c_1 = 2,\ c_\sigma = 6,\ \eta = \frac{1}{4}$ gave a satisfactory balance between runtime and accuracy.

4.4.1. *Accuracy.* In Figure 4 we plot the average EMD(1) error as a function of the noise level σ. The EMD(ω) error was zero for all but one of the trials. From the figure it is clear that the multiscale algorithm performs well in the presence of noise, even with c_1 as small as 2. This is in contrast to the rounding algorithm of Section 3, which required $p > (\sigma/\varepsilon)^{2/3}$ to achieve a similar accuracy. The multiscale error correction allows us to take much coarser sampling rates to achieve a given error. As we show in the next subsection, these coarser sampling rates lead to much improved runtime.

4.4.2. *Runtime.* In Figure 5 (A) we plot the average runtime (in CPU ticks) as a function of the sparsity k. From the figure we can see that the runtime scales slightly superlinearly with k, which is expected given the runtime bound $\mathrm{O}(k \log(k) \log(N/k))$ of Subsection 4.3.1. Moreover, we note that for all levels of sparsity tested, the multiscale algorithm outperforms FFTW (which, as noted in Subsection 3.2, takes approximately 10^9 CPU ticks on the same length signal).

Algorithm 1 MULTISCALEFREQEST

Input: $S(t), N, k, p, \sigma, c_\sigma, \eta$

Output: $\{\widetilde{\omega}_\ell\}_{\ell=1}^{\widetilde{k}}$

$\tau \leftarrow \frac{c_\sigma \sigma}{\sqrt{p}},\ \beta \leftarrow \frac{\sqrt{p}}{2c_\sigma \sigma},\ m \leftarrow 1 + \left\lfloor \log_\beta \frac{N}{p} \right\rfloor,\ \text{vote}_\ell \leftarrow 0,\ \ell = 1, \ldots, k$

$\widehat{\mathbf{S}}_p \leftarrow$ FFT of $\frac{1}{p}$-samples of $S(t)$

for $j = 0$ to m **do**

  $\varepsilon_j \leftarrow \beta^j / 2N$

  $\widehat{\mathbf{S}}_{p,\varepsilon_j} \leftarrow$ FFT of ε_j-shifted $\frac{1}{p}$-samples of $S(t)$

  for $\ell = 1$ to k **do**

    $h \leftarrow$ index of ℓ^{th} largest peak in $\widehat{\mathbf{S}}_p$

    $r \leftarrow \left| \frac{|\widehat{\mathbf{S}}_{p,\varepsilon_j}[h]|}{|\widehat{\mathbf{S}}_p[h]|} - 1 \right|$

    if $r > \tau$ **then**

      $\text{vote}_\ell \leftarrow \text{vote}_\ell + 1$

    end if

    $b_j \leftarrow \frac{1}{2\pi} \operatorname{Arg}\left(\frac{\widehat{\mathbf{S}}^{\text{n}}_{p,\varepsilon_j}[h]}{\widehat{\mathbf{S}}^{\text{n}}_p[h]} \right)$

    if $j = 0$ **then**

      $\widetilde{\omega}_\ell \leftarrow b_j / \varepsilon_j$

    else

      $\widetilde{\omega}_\ell \leftarrow \widetilde{\omega}_\ell + (b_j - \varepsilon_j \widetilde{\omega}_\ell) \pmod{[-\frac{1}{2}, \frac{1}{2})} / \varepsilon_j$

    end if

    if $j = m$ **then**

      $\widetilde{\omega}_\ell \leftarrow p \cdot \text{round}\left(\frac{\widetilde{\omega}_\ell - h}{p} \right) + h$

    end if

  end for

end for

return $\widetilde{\omega}_\ell$ with $\text{vote}_\ell \leq \eta(m+1)$

In Figure 5 (B) we plot the average runtime as a function of the noise level σ. Only a very mild dependence of the runtime on σ is visible for $\sigma \leq 0.128$. The slight increase for larger σ is due to the requirement $\beta > 1$, as noted in Subsection 4.3.2. Note the contrast with Figure 3 (B) for the rounding algorithm, where the dependence was of the form $\sigma^{2/3}$ as required by the choice of sampling rate p.

5. Conclusion

In this paper we gave two extensions of the sparse Fourier algorithm of [LWC13] to handle noisy signals. The first of these was a minor modification of the original algorithm that

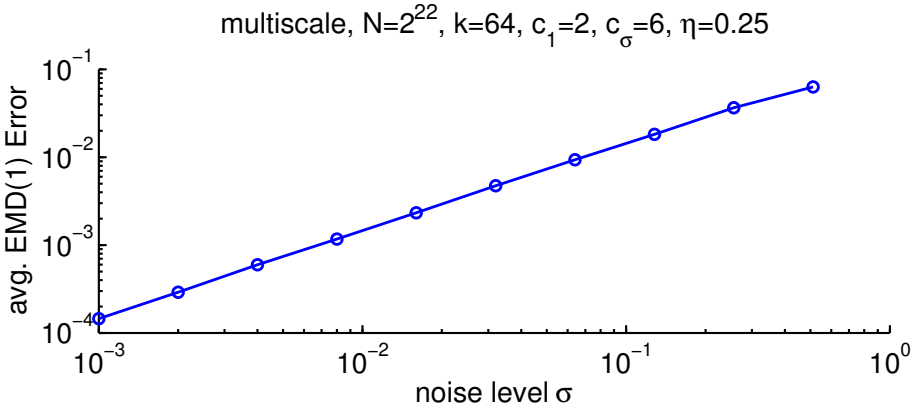


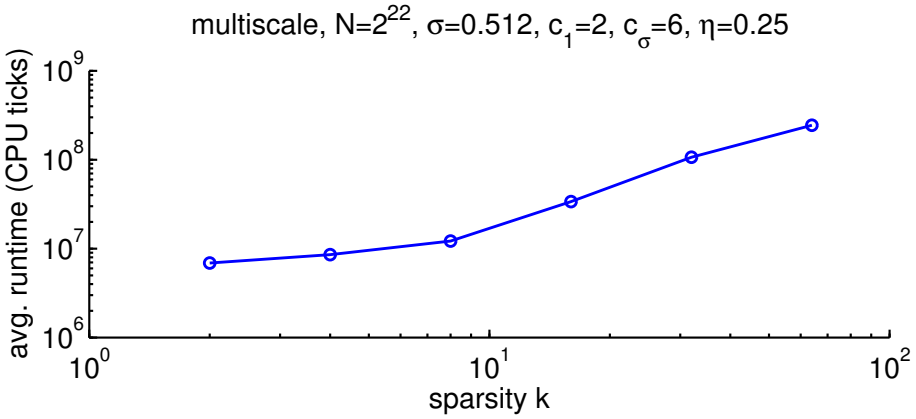
FIGURE 4. Average EMD(1) error vs. noise level σ for the algorithm with multiscale frequency identification.

involved rounding frequency estimates to the nearest integer with the correct residue modulo the sampling rate. We showed that in order for this modification to correctly identify the true frequencies in Gaussian noise of standard deviation σ the sampling rate needed to satisfy $p \geq \sigma^{2/3}$. While this resulted in accurate approximations of the Fourier transform in the EMD(1) metric, the sampling rate requirement forced the algorithms to be slow in practice.

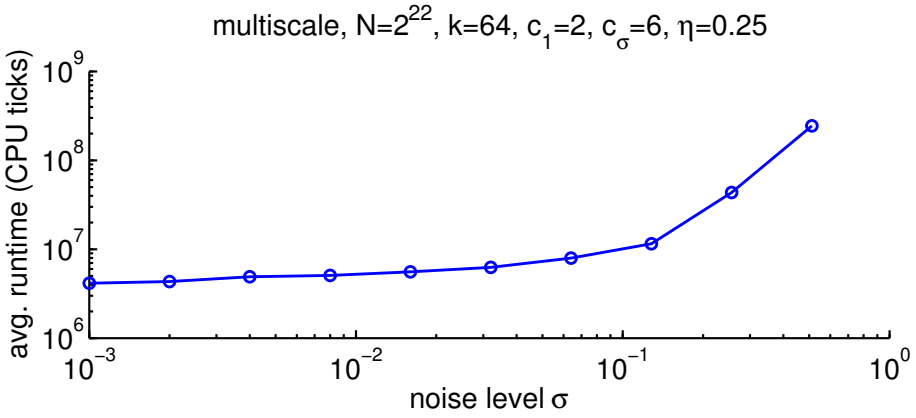
The second extension overcame this pitfall by introducing a novel multiscale approach to frequency estimation in the sparse Fourier transform context. By using samples of the input at multiple time shifts spaced geometrically, our algorithm exhibits a form of error correction in its frequency estimation. This allows the use of much coarser sampling rates than the first modification, which in turn leads to greatly reduced runtimes in our empirical evaluation. The error correction of our multiscale algorithm is to the best of our knowledge novel in the sparse Fourier transform context, and we believe it is a promising approach for further investigation.

ACKNOWLEDGMENTS

During the preparation of this manuscript we became aware of similar work by Laurent Demanet and his collaborators. We kindly acknowledge their generosity in sharing their work with us.



(A)



(B)

FIGURE 5. (A) Average runtime (in CPU ticks) vs. sparsity k for the algorithm with multiscale frequency identification. (B) Average runtime vs. noise level σ.

REFERENCES

[AD96] C. Anderson and M. D. Dahleh, *Rapid computation of the discrete Fourier transform*, SIAM J. Sci. Comput. **17** (1996), no. 4, 913–919.

[Aka10] A. Akavia, *Deterministic Sparse Fourier Approximation via Fooling Arithmetic Progressions*, Conference on Learning Theory (CoLT), 2010.

[CT65] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp. **19** (1965), 297–301.

[DDGV06] I. Daubechies, R. DeVore, C. S. Gunturk, and V. Vaishampayan, *A/D conversion with imperfect quantizers*, IEEE Transactions on Information Theory **52** (2006), no. 3, 874–885.

[FJ05] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, Proceedings of the IEEE **93** (2005), no. 2, 216–231, Special issue on "Program Generation, Optimization, and Platform Adaptation".

[GGI+02] A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, *Near-optimal sparse Fourier representations via sampling*, Symposium on Theory of Computing, 2002, pp. 152–161.

[GMS05] A. Gilbert, S. Muthukrishnan, and M. Strauss, *Improved time bounds for near-optimal sparse Fourier representations*, SPIE Wavelets XI, 2005.

[HIKP12a] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, *Nearly optimal sparse Fourier transform*, Proceedings of the 44th symposium on Theory of Computing, ACM, 2012, pp. 563–578.

[HIKP12b] ———, *Simple and practical algorithm for sparse Fourier transform*, Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2012, pp. 1183–1194.

[IGS07] M. Iwen, A. Gilbert, and M. Strauss, *Empirical evaluation of a sub-linear time sparse DFT algorithm*, Commun. Math. Sci. **5** (2007), no. 4, 981–998.

[Iwe10] M. Iwen, *Combinatorial sublinear-time Fourier algorithms*, Found. Comput. Math. **10** (2010), no. 3, 303–338.

[LWC13] D. Lawlor, Y. Wang, and A. Christlieb, *Adaptive sub-linear time Fourier algorithms*, to appear in Advances in Adaptive Data Analysis (2013), preprint available at arXiv:1207.6368.

[RTG00] Y. Rubner, C. Tomasi, and L. Guibas, *The earth mover's distance as a metric for image retrieval*, International Journal of Computer Vision **40** (2000), no. 2, 99–121.

Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA.

E-mail address: christlieb@math.msu.edu

Statistical and Applied Mathematical Sciences Institute, Department of Mathematics, Duke University, Durham, NC 27708, USA.

E-mail address: djl@math.duke.edu

Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA.

E-mail address: ywang@math.msu.edu